

TRƯỜNG ĐẠI HỌC CẦN THƠ
TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG



Niên luận Cơ sở ngành Kỹ thuật phần mềm
TRÒ CHƠI ĐÁNH MÁY TỐC ĐỘ
CRABTYPE

Cán bộ hướng dẫn:
Gv. Trương Thị Thanh Tuyền
MSCB: 1068

Sinh viên thực hiện:
Quách Minh Hón
MSSV: B2110078
Lớp: DI2196A1

Học kỳ 2, năm học 2023-2024

ĐÁNH GIÁ KẾT QUẢ THỰC HIỆN NIÊN LUẬN CƠ SỞ NGÀNH KTPM

(Học kỳ : 02, Năm học 2023 - 2024)

TÊN ĐỀ TÀI: Trò chơi đánh máy tốc độ CrabType

GIÁO VIÊN HƯỚNG DẪN:

STT	HỌ VÀ TÊN	MSCB
1	Trương Thị Thanh Tuyền	1068

SINH VIÊN THỰC HIỆN:

HỌ VÀ TÊN	MSSV	THƯỜNG (Tối đa 1,0 điểm)	ĐIỂM (thang điểm 10)
Quách Minh Hớn	B2110078		

I. HÌNH THỨC (0,5 điểm)	
Bìa (tối đa 0,25 điểm)	
<ul style="list-style-type: none">▪ Đầy đủ các thông tin▪ Đúng định dạng	
Bố cục (tối đa 0,25 điểm)	
<ul style="list-style-type: none">▪ Trang đánh giá kết quả thực hiện niên luận 1▪ Mục lục: cấu trúc chương, mục và tiểu mục▪ Phụ lục (nếu có)▪ Tài liệu tham khảo	
II. NỘI DUNG (3,5 điểm)	
Giới thiệu (tối đa 0,5 điểm)	
<ul style="list-style-type: none">▪ Mô tả bài toán (0,25 điểm)▪ Mục tiêu cần đạt, hướng giải quyết (0,25 điểm)	
Lý thuyết (tối đa 0,5 điểm)	
<ul style="list-style-type: none">▪ Các khái niệm sử dụng trong chương trình (0,25 điểm)▪ Kết quả vận dụng lý thuyết trong đề tài (0,25 điểm)	
Ứng dụng (tối đa 2,0 điểm)	
<ul style="list-style-type: none">▪ Phân tích yêu cầu, xây dựng các cấu trúc dữ liệu (0,5 điểm)▪ Sơ đồ chức năng, lưu đồ giải thuật giải quyết vấn đề (1,0 điểm)▪ Giới thiệu sử dụng chương trình (0,5 điểm)	
Kết luận (tối đa 0,5 điểm)	
<ul style="list-style-type: none">▪ Nhận xét kết quả đạt được▪ Hạn chế▪ Hướng phát triển	
III. CHƯƠNG TRÌNH DEMO (5,0 điểm)	
Giao diện thân thiện với người dùng (1,0 điểm)	
Hướng dẫn sử dụng (0,5 điểm)	
Kết quả thực hiện đúng với kết quả của phần ứng dụng (tối đa 3,5 điểm)	
<ul style="list-style-type: none">▪ Kết quả đúng (2,0 điểm)▪ Cách thức thực hiện hợp lý (1,0 điểm)▪ Chức năng bổ sung, sáng tạo (0,5 điểm)	

Cần Thơ, ngày tháng năm 2024
GIÁO VIÊN CHẤM

Trương Thị Thanh Tuyền

Lời cảm ơn



Sau hơn vài tháng học tập và nghiên cứu đề tài “Trò chơi đánh máy tốc độ CrabType” em đã không ngừng củng cố, học thêm được nhiều kiến thức và mở rộng sự hiểu biết về lĩnh vực Công nghệ Thông tin qua sự dạy dỗ, đào tạo của các thầy cô bộ môn trong khoa kỹ thuật phần mềm.

Thông qua báo cáo này, em xin được gửi lời cảm ơn chân thành đến Ban Giám Hiệu Trường và các thầy cô trong khoa đã tạo điều kiện tốt nhất cho em trong suốt khoảng thời gian học tập và sinh hoạt tại trường. Em cũng xin chân thành cảm ơn cô Trương Thị Thanh Tuyền, giáo viên phụ trách hướng dẫn em tận tình trong suốt quá trình thực hiện đề tài.

Em xin chân thành cảm ơn quý thầy cô đã nhiệt tình giảng dạy, truyền đạt những kiến thức, kinh nghiệm quý báu và hơn hết là tình cảm mà quý thầy cô đã dành cho em trong suốt quá trình giảng dạy.

Mặc dù em đã cố gắng hoàn thành niên luận cơ sở trong phạm vi và khả năng cho phép nhưng chắc chắn sẽ không tránh khỏi những thiếu sót và hạn chế. Em rất mong nhận được sự đóng góp ý kiến quý báu của quý Thầy Cô và các bạn để đề tài niên luận này ngày càng hoàn thiện hơn.

Cần Thơ, ngày 14 tháng 05 năm 2024

Người viết

Quách Minh Hón

Mục lục

ĐÁNH GIÁ KẾT QUẢ THỰC HIỆN NIÊN LUẬN CƠ SỞ NGÀNH KTPM	1
Lời cảm ơn	2
Mục lục.....	3
Danh mục ảnh.....	5
PHẦN 1: GIỚI THIỆU.....	6
1. Mô tả bài toán	6
2. Mục tiêu cần đạt được	7
3. Hướng giải quyết.....	7
4. Kế hoạch thực hiện	8
PHẦN 2: LÝ THUYẾT CƠ SỞ	9
1. Cơ sở lý thuyết	9
1.1. Thuật toán Levenshtein Distance (Edit Distance).....	9
1.2. Thuật toán xáo trộn Knuth (Fisher – Yates).....	10
1.3. React.js.....	10
1.4. Express.js	10
1.5. Tailwind CSS	11
1.6. MySQL.....	11
2. Kết quả vận dụng.....	11
PHẦN 3: ỨNG DỤNG	12
1. Phân tích yêu cầu	12
1.1. Yêu cầu chức năng	12
1.2. Yêu cầu phi chức năng	12
2. Thiết kế dữ liệu	13
2.1. Sơ đồ CDM	13
2.2. Bảng Account.....	13
2.3. Bảng Mode.....	13
2.4. Bảng Config	13
2.5. Bảng Vocab_difficulty	14
2.6. Bảng Record.....	14
3. Sơ đồ và mô tả chức năng	14
3.1. Sơ đồ Use Case	14
3.2. Sơ đồ phân rã chức năng	15
3.3. Xác thực tài khoản	15
3.4. Xem thống kê kết quả	16
3.5. Chọn chế độ	16
3.6. Chọn config.....	17

3.7.	Chọn độ khó từ vừng.....	17
3.8.	Bắt đầu lượt chơi.....	18
3.9.	Reset lượt chơi	18
3.10.	Xem kết quả	18
3.11.	Hiển thị thống kê kết quả	19
3.12.	Kiểm tra CapLock.....	19
3.13.	Tính toán kết quả.....	20
3.14.	Hiển thị kết quả.....	20
4.	Lưu đồ giải thuật	21
4.1.	Thuật toán xáo trộn Knuth	21
4.2.	Thuật toán Levenshtein Distance	22
5.	Giới thiệu chương trình.....	23
5.1.	Trang chủ	23
5.2.	Trang cá nhân / tài khoản.....	25
5.3.	Trang xác thực.....	26
	PHẦN 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	27
1.	Kết quả đạt được	27
2.	Hạn chế.....	27
3.	Hướng phát triển	27
	TÀI LIỆU THAM KHẢO	28

Danh mục ảnh

Hình 1 - Sơ đồ CDM	13
Hình 2 - Sơ đồ Use Case tổng quan	14
Hình 3 - Sơ đồ phân rã chức năng.....	15
Hình 4 - Lưu đồ giải thuật xáo trộn Knuth.....	21
Hình 5 - Lưu đồ giải thuật Levenshtein Distance	22
Hình 6 - Trang chủ	23
Hình 7 - Trang chủ sau khi xác thực tài khoản	23
Hình 8 - Trang chủ (chế độ quote)	24
Hình 9 - Trang chủ (chế độ time).....	24
Hình 10 - Trang chủ (chế độ random).....	24
Hình 11 - Trang cá nhân / tài khoản.....	25
Hình 12 - Trang xác thực	26

PHẦN 1: GIỚI THIỆU

1. Mô tả bài toán

Trong tình hình công nghệ phát triển nhanh chóng, kỹ năng đánh máy nhanh lẫn kỹ năng tiếng Anh thực sự là các yếu tố không thể thiếu khi xin việc. CrabType là một trò chơi đánh máy tốc độ trên web được lấy ý tưởng từ trang web game đánh máy MonkeyType, một nền tảng phổ biến trong cộng đồng yêu thích đánh máy tốc độ. Mục đích chính của trò chơi là giúp người chơi cải thiện tốc độ đánh máy cũng như phát triển kỹ năng tiếng Anh của họ, giúp họ mở rộng vốn từ vựng và ngữ pháp. CrabType được xây dựng trên Framework React.js dành cho front-end và Express.js dành cho back-end.

Quy luật của trò chơi rất đơn giản, người chơi đánh máy theo những ký tự hiện ra trên màn hình nhanh nhất có thể đến khi hết ký tự hoặc hết thời gian. Kết quả của người bao gồm WPM (Words Per Minute) và độ chính xác (Accuracy). Ví dụ: Kết quả trung bình trên thế giới là 45 WPM và 90% độ chính xác. Sau khi kết thúc lượt chơi, kết quả của người chơi sẽ được lưu vào một cơ sở dữ liệu trên MySQL nếu người chơi đã đăng nhập.

Trò chơi gồm 4 chế độ: time, words, quote, random.

- Time là chế độ tính thời gian, người chơi đánh chữ đến khi thời gian kết thúc, bao gồm các lựa chọn 15s, 30s, 60s, 120s.
- Words là chế độ đánh số từ, người chơi đánh hết số từ trên màn hình, bao gồm các lựa chọn 10 từ, 25 từ, 50 từ, 100 từ.
- Quote là chế độ đánh theo câu cho sẵn, người chơi đánh theo một câu có độ dài tùy chọn có trên màn hình, bao gồm các lựa chọn short (0 – 100 ký tự), medium (101 – 300 ký tự), long (301 – 600 ký tự), extended (trên 600 ký tự).
- Random là chế độ đánh ký tự ngẫu nhiên, website hiện giao diện bàn phím ảo, người chơi đánh ký tự được tô hồng trên bàn phím ảo. Chế độ này không có kết thúc, với mục đích cho người chơi làm quen với bàn phím.

Hiện tại, CrabType chỉ hỗ trợ ngôn ngữ Anh với 4 độ khó từ vựng: English, English 1k, English 5k, English 10k. Độ khó được phân loại dựa trên một số yếu tố bao gồm: Độ dài của từ, độ phổ biến của từ trong tiếng Anh giao tiếp, tần số xuất hiện của chữ cái, sự kết hợp của các chữ cái, độ phức tạp của từ vựng.

Giao diện của trang web được thiết kế một cách đơn giản nhằm tránh gây mất tập trung trong quá trình luyện tập. Khi truy cập website, người chơi có thể bắt đầu lượt chơi và xem kết quả trực tiếp mà không cần đăng ký tài khoản, điều này tạo trải nghiệm thân thiện đối với người chơi. Tuy nhiên, kết quả đó sẽ không được lưu trữ vào cơ sở dữ liệu. Để có thể theo dõi tiến độ của bản thân, người chơi cần đăng ký tài khoản hoặc đăng nhập vào hệ thống.

Độ chính xác được tính bằng tỷ lệ số ký tự đánh đúng trên tổng số ký tự đánh:

$$A = \frac{N}{S} \times 100$$

- A: Độ chính xác (%)
 - N: Số lượng ký tự được đánh chính xác
 - S: Tổng ký tự được đánh

WPM được tính bằng Gross WPM trừ các lỗi chưa được chỉnh sửa và chia cho thời gian hoàn thành. Trong đó, Gross WPM được tính bằng độ dài input của người chơi chia cho 5 (trung bình ký tự trong mỗi từ) và chia lần nữa cho thời gian hoàn thành.

$$Gross\ WPM = \frac{\left(\frac{n}{5}\right)}{t}$$

- n : Tổng ký tự người chơi đánh
- t : Thời gian hoàn thành (phút)

$$Net\ WPM = Gross\ WPM - \left(\frac{e}{t}\right)$$

$$= \frac{\left[\left(\frac{n}{5}\right) - e\right]}{t}$$

- e : Lỗi chưa được chỉnh sửa
- n : Tổng ký tự người chơi đánh
- t : Thời gian hoàn thành (phút)

2. Mục tiêu cần đạt được

- Giao diện đơn giản, tăng cường sự tập trung cho người chơi.
- Chức năng xác thực tài khoản đơn giản.
- Chức năng phát hiện CapsLock.
- Lưu kết quả của người chơi vào cơ sở dữ liệu.
- Truy xuất và thống kê các thông tin của người chơi.
- Áp dụng thuật toán Knuth Shuffle.
- Áp dụng thuật toán Levenshtein Distance để tính WPM.
- Thời gian phản hồi nhanh.
- Có 4 chế độ chơi:
 - o Time: 15s, 30s, 60s, 120s.
 - o Words: 10 từ, 25 từ, 50 từ, 100 từ.
 - o Quote: short, medium, long, extended.
 - o Random
- Có 4 độ khó từ vựng:
 - o English
 - o English 1k
 - o English 5k
 - o English 10k

3. Hướng giải quyết

Để giải quyết bài toán “Trò chơi đánh máy tốc độ”, cần tiến hành các bước sau đây:

- Tìm hiểu và nghiên cứu thuật toán xáo trộn Knuth và thuật toán Levenshtein Distance.
- Phân tích các yêu cầu: Xác định các yêu cầu tính năng website cần có: Xác thực tài khoản, chọn chế độ, chọn config, chọn độ khó từ vựng, bắt đầu lượt chơi, reset lượt chơi, xem kết quả, xem thống kê kết quả, hiển thị kết quả, hiển thị thống kê kết quả, kiểm tra CapsLock, tính toán kết quả.
- Xây dựng cấu trúc dữ liệu: bao gồm cơ sở dữ liệu để lưu tài khoản, lượt chơi và các cài đặt cho lượt chơi.

- Vẽ sơ đồ thực thể quan hệ CDM.
- Sơ đồ chức năng: Vẽ sơ đồ Usecase và cây phân rã chức năng để mô tả các chức năng chính của website.
- Xây dựng chương trình:
 - o Phác thảo các giao diện người dùng trên nền tảng React.js.
 - o Trang trí website với Tailwind CSS.
 - o Xây dựng các chức năng liên quan.
 - o Xây dựng các bảng trong cơ sở dữ liệu MySQL.
 - o Xây dựng các APIs với Express.js.
- Kiểm thử chương trình:
 - o Kiểm tra các thành phần giao diện.
 - o Kiểm tra các chức năng của website.
 - o Kiểm tra kết nối của Express.js với MySQL.
 - o Kiểm thử các APIs vừa xây dựng.
 - o Khắc phục lỗi trong quá trình kiểm thử.

4. Kế hoạch thực hiện

Tuần	Công việc thực hiện
2	Nhận đề tài
3 - 4	Phân tích đề tài
5	Nộp tài liệu đặc tả yêu cầu chi tiết bài toán
6	Tìm hiểu về thuật toán
7 - 9	Thiết kế giao diện và xây dựng các chức năng
10	Nộp tài liệu thiết kế giải thuật và dữ liệu sử dụng
11- 16	Kiểm thử và hoàn thiện website (tuần 15 demo)
17	Viết quyền báo cáo
18	Nộp quyền niên luận và báo cáo demo

PHẦN 2: LÝ THUYẾT CƠ SỞ

1. Cơ sở lý thuyết

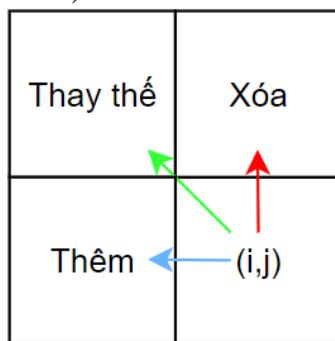
1.1. Thuật toán Levenshtein Distance (Edit Distance)

Thuật toán Levenshtein Distance hoặc còn gọi là Edit Distance Là thuật toán được sử dụng rộng rãi trong các phần mềm kiểm tra chính tả, kiểm tra đạo văn, tìm kiếm. Nó xác định số các khác biệt giữa 2 Strings. Số khác biệt đó được gọi là khoảng cách Levenshtein hoặc ngắn gọn là khoảng cách.

Khoảng cách Levenshtein là số lần ít nhất phải chỉnh sửa một ký tự (thêm, xóa, thay thế) để biến đổi một String thành String khác. Thuật toán sử dụng quy hoạch động để lưu số lần chỉnh sửa. Nó có độ phức tạp là $O(m*n)$ với m, n lần lượt là độ dài của String 1 và String 2.

Phương pháp hiện đại:

- Khởi tạo ma trận để lưu kết quả của các bài toán nhỏ, không quan trọng vị trí của 2 Strings, có nghĩa là String 1 có thể ở trên cùng, String 2 ở bên trái và ngược lại.
- Gán giá trị cho cột và hàng đầu tiên trong ma trận từ $0 \rightarrow n$ và $0 \rightarrow m$.
- Bắt đầu tính khoảng cách nhỏ nhất từ ô $(1,1) \rightarrow$ ô (n,m) theo quy tắc sau:
 - Có 3 giá trị trong 3 ô gần ô đang tính cần quan tâm:
 - Xóa (ô phía trên)
 - Thay thế (ô chéo bên trái)
 - Thêm (ô bên trái)



- Lấy giá trị nhỏ nhất trong 3 giá trị trên, nếu $String1[i] \neq String2[j]$ thì +1 vào giá trị nhỏ nhất.
- Lặp lại quy trình cho từng ô đến hết ma trận.
- Ô dưới cùng bên phải sẽ là Khoảng cách Levenshtein nhỏ nhất để chuyển String1 thành String2.

Một trong những ưu điểm của thuật toán Levenshtein Distance là khả năng xử lý các lỗi khác nhau. Điều này giúp nó linh hoạt trong việc xử lý dữ liệu trong các trường hợp thực tế với dữ liệu không hoàn hảo. Bên cạnh đó, do thuật toán tính toán khoảng cách nhỏ nhất để biến đổi một String thành String khác, điều này cung cấp một thước đo rõ ràng và có thể định lượng được về mức độ giống nhau của hai String.

1.2. Thuật toán xáo trộn Knuth (Fisher – Yates)

Thuật toán trộn Knuth hoặc còn có tên gọi cũ là Fisher – Yates là một thuật toán dùng để xáo trộn một chuỗi hữu hạn. Thuật toán này nhận vào một danh sách chứa tất cả các phần tử của chuỗi, và liên tục xác định phần tử tiếp theo trong chuỗi đã được trộn bằng cách ngẫu nhiên chọn một phần tử từ danh sách cho đến khi không còn phần tử nào còn lại. Thuật toán tạo ra một hoán vị không thiên vị: mỗi hoán vị đều có khả năng xảy ra như nhau. Phiên bản hiện đại của thuật toán có thời gian thực thi tỉ lệ thuận với số lượng phần tử đang được trộn và trộn chúng ngay tại chỗ.

Phương pháp hiện đại:

- Chọn phần tử $a[n-1]$, chọn ngẫu nhiên x từ $0 \rightarrow n-1$ và hoán đổi $a[n-1]$ với $a[x]$.
- Chọn phần tử $a[n-2]$, chọn ngẫu nhiên x từ $0 \rightarrow n-2$ và hoán đổi $a[n-2]$ với $a[x]$.
- Sau khi lặp lại $n-1$ lần thì array sẽ được xáo trộn.

Thuật toán Knuth thường được sử dụng nhiều hơn các thuật toán xáo trộn khác bởi vì những ưu điểm của nó. Một trong số đó là hiệu suất của thuật toán, nó có độ phức tạp $O(n)$ nên nó rất hiệu quả khi xử lý nhiều phần tử. Ngoài ra, nó còn có độ ngẫu nhiên cao, bất kỳ phần tử nào cũng đều có xác suất được chọn như nhau. Không chỉ vậy, thuật toán không cần thêm bộ nhớ phụ để chứa các phần tử, điều này góp phần giảm thiểu tài nguyên thuật toán sử dụng.

1.3. React.js

React.js, hay đơn giản là React, là một framework JavaScript mạnh mẽ với mục đích xây dựng giao diện người dùng (UI) cho các ứng dụng web. Được phát triển bởi Facebook và ra mắt vào năm 2013, React đã trở thành một trong những thư viện phát triển front-end phổ biến nhất nhờ kiến trúc dựa trên thành phần, DOM ảo (Virtual DOM) và quy trình vẽ hiệu quả.

Một trong những điểm mạnh của React.js là kiến trúc React sử dụng. Đó là kiến trúc Component-based, nó tập trung vào việc xây dựng các thành phần UI độc lập và có thể tái sử dụng, giúp cho việc phát triển và bảo trì mã nguồn trở nên dễ dàng hơn.

Ngoài ra, React sử dụng một biến thể của DOM được gọi là DOM ảo (Virtual DOM) để tối ưu hóa hiệu suất của website. Thay vì cập nhật toàn bộ DOM mỗi khi có thay đổi, React chỉ cập nhật các thành phần DOM cần thiết, góp phần làm giảm thời gian phản hồi và tăng hiệu suất của ứng dụng.

1.4. Express.js

Express.js là một framework web backend mạnh mẽ được sử dụng để tạo ra các ứng dụng web và API. Nó được xây dựng trên cơ sở các module của Node.js. Ra mắt lần đầu vào năm 2010, Express nhanh chóng trở thành một trong những công cụ phát triển backend phổ biến nhất cho JavaScript.

Express.js cung cấp một cách tiếp cận đơn giản và tối giản để xử lý các yêu cầu HTTP, quản lý các route (đường dẫn URL) và xây dựng các ứng dụng web có hiệu suất cao. Nó cho phép xây dựng các ứng dụng web phức tạp hoặc các API RESTful một cách nhanh chóng và dễ dàng.

1.5. Tailwind CSS

Tailwind CSS là một thư viện CSS được thiết kế để giúp người dùng xây dựng giao diện (UI) một cách nhanh chóng và hiệu quả. Khác với các thư viện CSS truyền thống như Bootstrap, Tailwind không cung cấp các thành phần (component) được định nghĩa trước, nó cung cấp các lớp ở dạng cơ bản để tạo ra các thiết kế tùy ý.

Với Tailwind CSS, người dùng có thể sử dụng các lớp CSS trực tiếp trên HTML để định dạng và trang trí các thành phần UI. Điều này giúp giảm bớt sự phức tạp của CSS, để người dùng có thể tập trung vào việc thiết kế. Qua đó, thúc đẩy tốc độ phát triển nhanh chóng.

1.6. MySQL

MySQL là một hệ quản trị cơ sở dữ liệu phổ biến, được sử dụng rộng rãi trong các ứng dụng web và doanh nghiệp. Được phát triển, duy trì và hỗ trợ bởi Oracle, MySQL cung cấp nhiều các tính năng và công cụ để lưu trữ, truy xuất và quản lý dữ liệu.

Một số ưu điểm của MySQL bao gồm khả năng mở rộng linh hoạt, hiệu suất cao, độ ổn định và tính linh hoạt trong việc tích hợp với nhiều ngôn ngữ lập trình và framework. MySQL cũng hỗ trợ các tính năng như giao dịch, bảo mật dữ liệu và sao lưu phục hồi, làm cho nó trở thành một lựa chọn phổ biến cho các dự án web và doanh nghiệp.

2. Kết quả vận dụng

- Áp dụng thuật toán Levenshtein Distance trong việc tính toán WPM.
- Áp dụng thuật toán xáo trộn Knuth trong việc chọn ngẫu nhiên các từ vựng.
- Áp dụng framework React.js trong đề tài để thiết kế giao diện người, nó giúp tăng tốc độ phát triển nhờ vào kiến trúc tái sử dụng thành phần. Do React.js sử dụng DOM ảo, nó góp phần giảm tối thiểu thời gian phản hồi và tăng hiệu suất của website.
- Áp dụng framework Express.js trong đề tài để xây dựng các APIs đơn giản giao tiếp với cơ sở dữ liệu. Express.js là một framework nhẹ và đơn giản, do đó, nó xử lý các yêu cầu từ client và xử lý dữ liệu từ server đến client một cách hiệu quả.
- Áp dụng thư viện Tailwind CSS để trang trí các thành phần của website.
- Áp dụng MySQL làm cơ sở dữ liệu của website. Bao gồm các bảng: Account, Mode, Config, Vocab_difficulty, Record.

PHẦN 3: ỨNG DỤNG

1. Phân tích yêu cầu

1.1. Yêu cầu chức năng

Các yêu cầu chức năng của website bao gồm:

- **Xác thực tài khoản:** Người dùng xác thực tài khoản bằng email và mật khẩu. Nếu chưa có tài khoản, người dùng có thể đăng ký với các thông tin sau: Username, email, mật khẩu.
- **Kiểm tra CapLock:** Hệ thống sẽ kiểm tra nếu nút CapLock trên bàn phím đang hoạt động.
- **Chọn chế độ:** Người dùng chọn chế độ để chơi bao gồm: Time, words, quote, random. Chế độ mặc định sẽ là words.
- **Chọn config:** Sau khi chọn chế độ chơi, người dùng có thể chọn các lựa chọn cho chế độ đó. Mặc định sẽ là lựa chọn đầu tiên của chế độ.
- **Chọn độ khó từ vựng:** Người dùng chọn độ khó của từ vựng trong tiếng Anh. Bao gồm: English, English 1k, English 5k, English 10k.
- **Bắt đầu lượt chơi:** Người dùng đánh phím bất kỳ để bắt đầu lượt chơi, lượt chơi kết thúc khi người dùng đánh hết ký tự trên màn hình hoặc hết thời gian.
- **Reset lượt chơi:** Người dùng bấm nút “Reset” để thực hiện lại hoặc bắt đầu lượt chơi mới. Chế độ và config của lượt chơi mới sẽ như lượt chơi trước.
- **Xem kết quả:** Ngay sau khi hoàn thành lượt chơi, người chơi có thể xem kết quả của lượt chơi đó. Bao gồm: WPM, độ chính xác, biểu đồ đường thể hiện WPM mỗi giây, độ khó từ vựng, chế độ, config, số ký tự đánh, thời gian đánh.
- **Tính toán kết quả:** Hệ thống sẽ tính toán WPM (Words Per Minute), độ chính xác (accuracy) và WPM mỗi giây.
- **Hiển thị kết quả:** Hệ thống hiển thị kết quả ngay sau khi kết thúc lượt chơi. Các thông tin được hiển thị bao gồm: WPM, độ chính xác, biểu đồ đường thể hiện WPM mỗi giây, độ khó từ vựng, chế độ, config, số ký tự đánh, thời gian đánh.
- **Xem thống kê kết quả:** Người dùng vào trang cá nhân để xem các thông tin sau: Số lượt chơi hoàn thành, WPM trung bình, tổng thời gian chơi, các lượt chơi tốt nhất của từng chế độ và config, biểu đồ cột thống kê các WPM, bảng lịch sử lượt chơi.
- **Hiển thị thống kê kết quả:** Hệ thống truy xuất và tính toán các thông tin từ cơ sở dữ liệu MySQL, sau đó hiển thị các thông tin đó ở trang cá nhân.

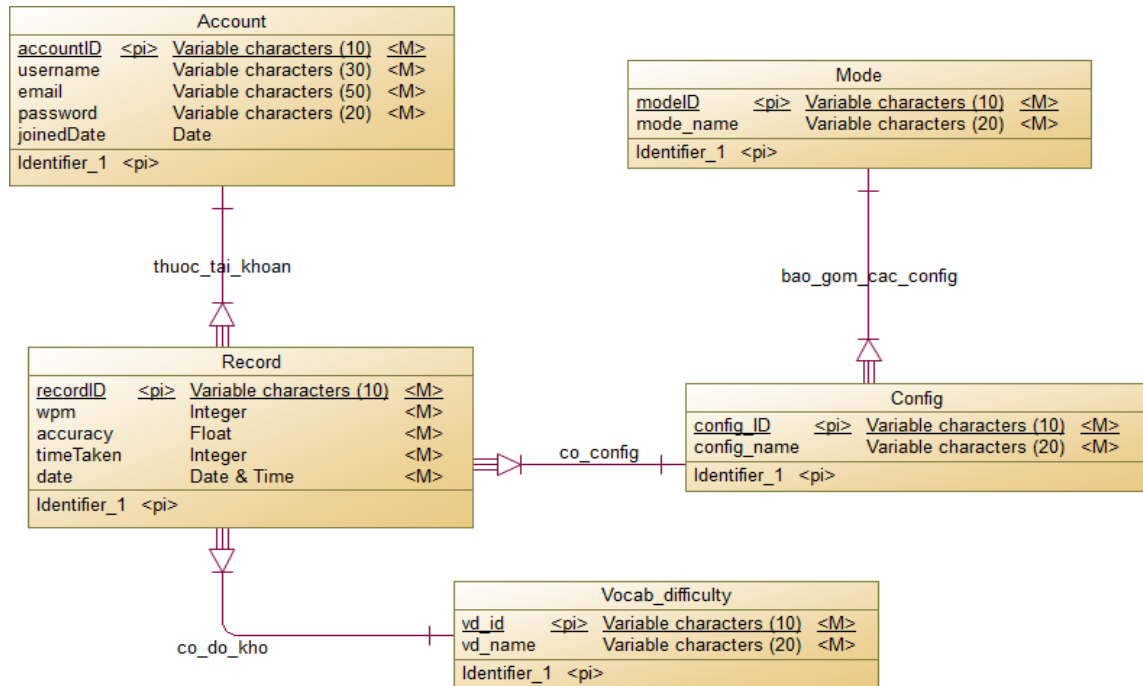
1.2. Yêu cầu phi chức năng

Các yêu cầu phi chức năng của website bao gồm:

- **Giao diện thân thiện với người dùng:** Giao diện của website được thiết kế đơn giản, dễ sử dụng với mục đích giúp người chơi tập trung.
- **Tốc độ xử lý nhanh:** Ứng dụng cần có thời gian phản hồi thấp vì người chơi thường bắt đầu và lặp lại lượt chơi nhiều lần liên tiếp.
- **Tính đầy đủ:** Phần mềm đáp ứng đầy đủ các yêu cầu chức năng đã được xác định và hoạt động chính xác, hiệu quả.

2. Thiết kế dữ liệu

2.1. Sơ đồ CDM



Hình 1 - Sơ đồ CDM

2.2. Bảng Account

TT	Thuộc tính	Kiểu dữ liệu	Kích thước	Khóa chính	Duy nhất	NOT NULL	Khóa ngoại	Mô tả
1	account_id	Varchar	10	x	x	x		ID tài khoản
2	Username	Varchar	30			x		Tên người dùng
3	Email	Varchar	50		x	x		Email người dùng
4	Password	Varchar	20			x		Mật khẩu tài khoản
5	joinedDate	Date				x		Ngày tạo tài khoản

2.3. Bảng Mode

TT	Thuộc tính	Kiểu dữ liệu	Kích thước	Khóa chính	Duy nhất	NOT NULL	Khóa ngoại	Mô tả
1	mode_id	Varchar	10	x	x	x		ID chế độ
2	mode_name	Varchar	20		x	x		Tên chế độ

2.4. Bảng Config

TT	Thuộc tính	Kiểu dữ liệu	Kích thước	Khóa chính	Duy nhất	NOT NULL	Khóa ngoại	Mô tả
1	config_id	Varchar	10	x	x	x		ID config
2	config_name	Varchar	20		x	x		Tên config
3	mode_id	Varchar	10			x	mode(mode_id)	ID chế độ

2.5. Bảng Vocab_difficulty

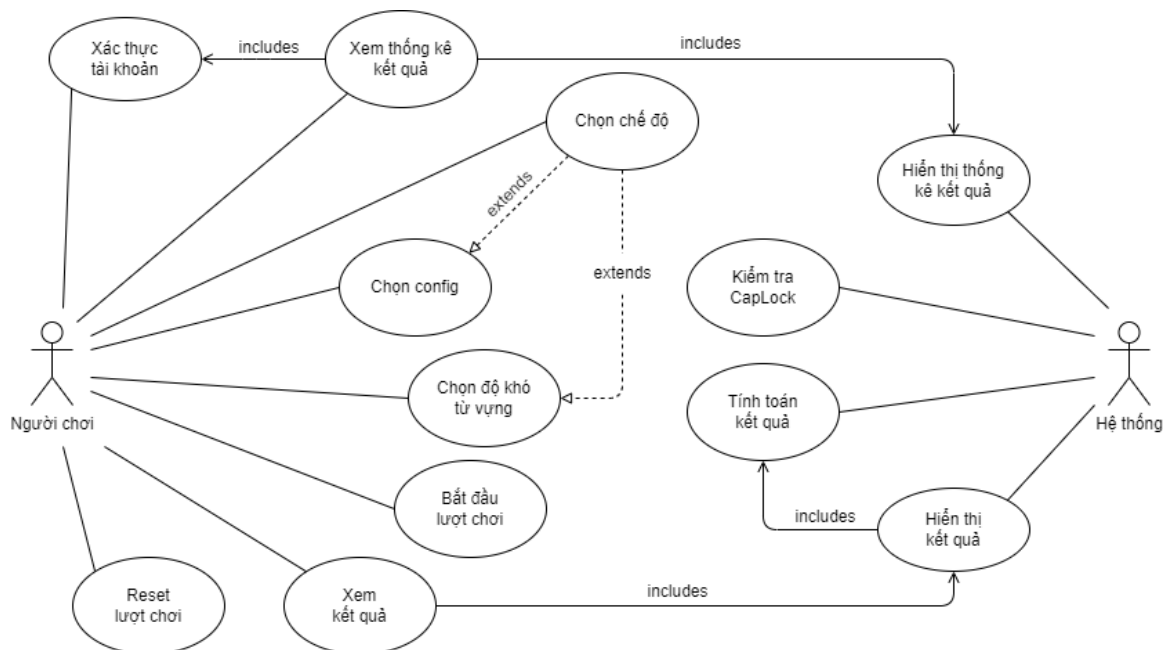
TT	Thuộc tính	Kiểu dữ liệu	Kích thước	Khóa chính	Duy nhất	NOT NULL	Khóa ngoại	Mô tả
1	vd_ID	Varchar	10	x	x	x		ID độ khó từ vựng
2	vd_name	Varchar	20		x	x		Tên độ khó

2.6. Bảng Record

TT	Thuộc tính	Kiểu dữ liệu	Kích thước	Khóa chính	Duy nhất	NOT NULL	Khóa ngoại	Mô tả
1	record_id	Varchar	15	x	x	x		ID lượt chơi
2	account_id	Varchar	10	x	x	x	account(account_id)	ID tài khoản
3	mode_id	Varchar	10	x	x	x	mode(mode_id)	ID chế độ
4	config_id	Varchar	10	x	x	x	config(config_id)	ID config
5	vd_id	Varchar	10	x	x	x	vocab_difficulty(vd_id)	ID độ khó từ vựng
6	wpm	Int				x		Số từ trên phút
7	accuracy	Float				x		Độ chính xác
8	timeTaken	Float				x		Thời gian hoàn thành lượt chơi
9	date	Date				x		Ngày lưu kết quả

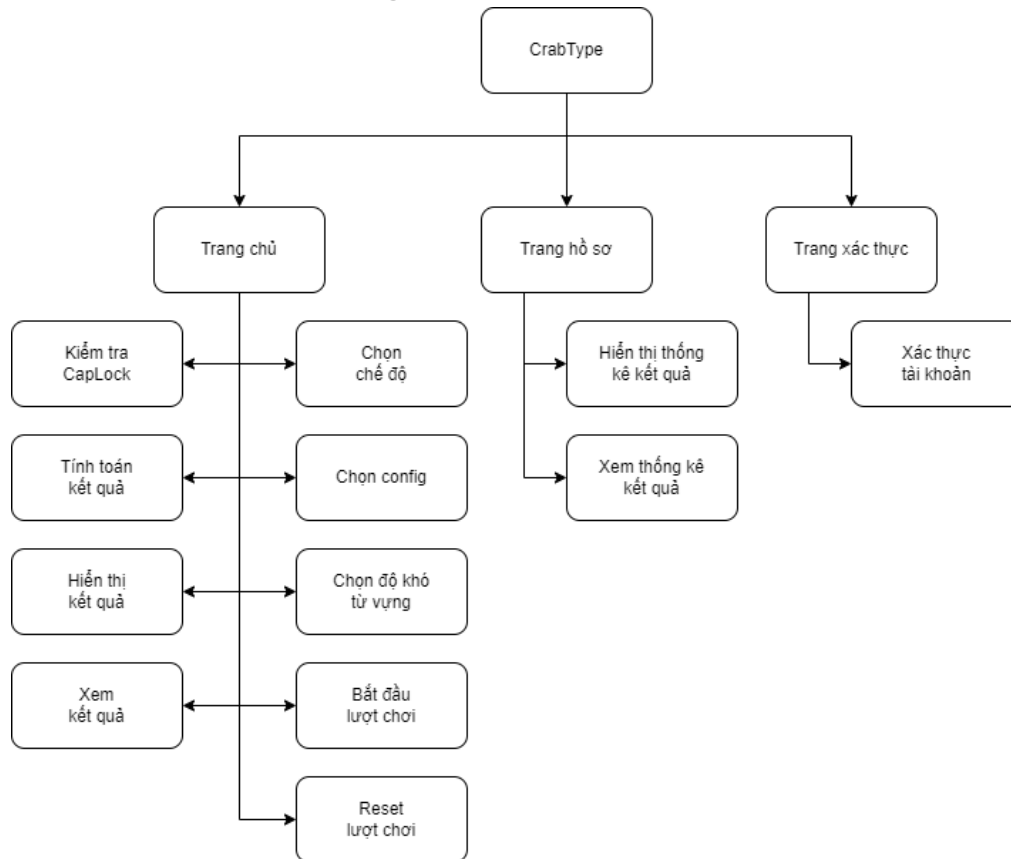
3. Sơ đồ và mô tả chức năng

3.1. Sơ đồ Use Case



Hình 2 - Sơ đồ Use Case tổng quan

3.2. Sơ đồ phân rã chức năng



Hình 3 - Sơ đồ phân rã chức năng

3.3. Xác thực tài khoản

Tên Use case: Xác thực tài khoản	
Actor chính: Người chơi	Mức độ cần thiết: Mong muốn
	Phân loại: Trung bình
Mô tả tóm tắt: Người chơi muốn đăng ký tài khoản hoặc đăng nhập vào website.	
Các mối quan hệ: + Association (kết hợp): Người chơi.	
Luồng xử lý bình thường của sự kiện: <ol style="list-style-type: none"> 1. Mở chức năng “Xác thực tài khoản” của website. 2. Người chơi được chuyển đến trang xác thực. 3. Trang xác thực hiển thị form đăng ký và form đăng nhập. 4. Người chơi chọn đăng ký hoặc đăng nhập. 	
Luồng luân phiên/đặc biệt (Alternate/Exceptional flows): <ol style="list-style-type: none"> A. Người chơi chọn đăng ký: <ol style="list-style-type: none"> 5. Nhập thông tin theo yêu cầu: Username, email, mật khẩu, nhập lại mật khẩu. 6. Click “Sign up” để gửi yêu cầu đăng ký và hệ thống tiến hành kiểm tra thông tin người chơi vừa nhập. <ol style="list-style-type: none"> i. Nếu email đã được sử dụng bởi tài khoản khác thì hệ thống sẽ thông báo “Account already exists” – Tài khoản đã tồn tại. ii. Nếu mật khẩu nhập lại không đúng, hệ thống sẽ thông báo “Verify password failed” – Mật khẩu không khớp. 7. Nếu thành công, người chơi được đưa về trang chủ. 	

B. Người chơi chọn đăng nhập:

5. Nhập thông tin theo yêu cầu: Email, mật khẩu.
6. Click “Sign in” để gửi yêu cầu đăng nhập và hệ thống tiến hành kiểm tra thông tin người chơi vừa nhập.
7. Nếu có thông tin sai, hệ thống thông báo “Wrong email or password” – Sai email hoặc mật khẩu.
8. Nếu thành công, người chơi được đưa về trang chủ.

3.4. Xem thống kê kết quả

Tên Use case: Xem thống kê kết quả	
Actor chính: Người chơi	Mức độ cần thiết: Bắt buộc
	Phân loại: Trung bình
Mô tả tóm tắt: Người chơi xem thống kê kết quả.	
Các mối quan hệ: + Association (kết hợp): Người chơi. + Includes (bao gồm): Xác thực tài khoản, Hiển thị thống kê kết quả	
Luồng xử lý bình thường của sự kiện: <ol style="list-style-type: none"> 1. Click vào trang tài khoản của website. 2. Người chơi được chuyển đến trang tài khoản. 3. Người chơi xem các thông tin về lịch sử lượt chơi của tài khoản. 	
Luồng luân phiên/đặc biệt (Alternate/Exceptional flows):	

3.5. Chọn chế độ

Tên Use case: Chọn chế độ	
Actor chính: Người chơi	Mức độ cần thiết: Bắt buộc
	Phân loại: Trung bình
Mô tả tóm tắt: Người chơi chế độ để chơi.	
Các mối quan hệ: + Association (kết hợp): Người chơi. + Extends (mở rộng): Chọn config, chọn độ khó từ vựng.	
Luồng xử lý bình thường của sự kiện: <ol style="list-style-type: none"> 1. Ở trang chủ, người chơi chọn 1 trong 4 chế độ. <ul style="list-style-type: none"> - Time - Words - Quote - Random 	
Luồng luân phiên/đặc biệt (Alternate/Exceptional flows): <ol style="list-style-type: none"> A. Người chơi chọn chế độ “Time”: <ol style="list-style-type: none"> 2. Hệ thống chọn config “15s” và độ khó “English” làm mặc định. 3. Hệ thống hiển thị đồng hồ đếm ngược với 15 giây. 4. Hệ thống sử dụng giải thuật xáo trộn Knuth để chọn các từ vựng. 5. Hệ thống hiển thị các từ vựng được chọn. B. Người chơi chọn chế độ “Words”: <ol style="list-style-type: none"> 2. Hệ thống chọn config “10 words” và độ khó “English” làm mặc định. 3. Hệ thống sử dụng giải thuật xáo trộn Knuth để chọn ra 10 từ vựng. 	

<p>4. Hệ thống hiển thị các từ vựng được chọn.</p> <p>C. Người chơi chọn chế độ “Quote”:</p> <p>2. Hệ thống chọn config “short” làm mặc định.</p> <p>3. Hệ thống chọn ngẫu nhiên quote trong danh sách “Short Quote”.</p> <p>4. Hệ thống hiển thị quote được chọn.</p> <p>D. Người chơi chọn chế độ “Random”:</p> <p>2. Hệ thống hiển thị bàn phím ảo trên màn hình.</p> <p>3. Hệ thống chọn một phím ngẫu nhiên để tô nền màu hồng.</p> <p>4. Phía dưới bàn phím hiển thị độ chính xác, số ký tự đúng / sai và nút reset.</p>
--

3.6. Chọn config

Tên Use case: Chọn config	
Actor chính: Người chơi	Mức độ cần thiết: Bắt buộc
	Phân loại: Trung bình
Mô tả tóm tắt: Người chơi bắt đầu lại lượt chơi.	
Các mối quan hệ: + Association (kết hợp): Người chơi.	
Luồng xử lý bình thường của sự kiện: <ol style="list-style-type: none"> 1. Các chế độ sẽ có các config khác nhau: <ul style="list-style-type: none"> - Time: 15s, 30s, 60s, 120s - Words: 10, 25, 50, 100 - Quote: short, medium, long, extended - Random: Không có config 	
Luồng luân phiên/đặc biệt (Alternate/Exceptional flows): <ol style="list-style-type: none"> A. Người chơi chọn chế độ “Time”: <ol style="list-style-type: none"> 2. Hệ thống hiển thị đồng hồ đếm ngược với n giây. 3. Hệ thống sử dụng giải thuật xáo trộn Knuth để chọn các từ vựng. 4. Hệ thống hiển thị các từ vựng được chọn. B. Người chơi chọn chế độ “Words”: <ol style="list-style-type: none"> 2. Hệ thống sử dụng giải thuật xáo trộn Knuth để chọn ra n từ vựng. 3. Hệ thống hiển thị các từ vựng được chọn. C. Người chơi chọn chế độ “Quote”: <ol style="list-style-type: none"> 2. Hệ thống chọn ngẫu nhiên quote trong danh sách quote dựa vào độ dài. 3. Hệ thống hiển thị quote được chọn. 	

3.7. Chọn độ khó từ vựng

Tên Use case: Chọn độ khó từ vựng	
Actor chính: Người chơi	Mức độ cần thiết: Bắt buộc
	Phân loại: Trung bình
Mô tả tóm tắt: Người chơi chọn độ khó từ.	
Các mối quan hệ: + Association (kết hợp): Người chơi.	
Luồng xử lý bình thường của sự kiện: <ol style="list-style-type: none"> 1. Ở trang chủ, người chơi chọn 1 trong 4 độ khó từ vựng: <ul style="list-style-type: none"> - English 	

<ul style="list-style-type: none"> - English 1k - English 5k - English 10k
2. Hệ thống sử dụng thuật toán xáo trộn Knuth để chọn các từ vựng tùy thuộc vào độ khó.
3. Hệ thống hiển thị các từ vựng được chọn.
Luồng luân phiên/đặc biệt (Alternate/Exceptional flows):

3.8. Bắt đầu lượt chơi

Tên Use case: Bắt đầu lượt chơi	
Actor chính: Người chơi	Mức độ cần thiết: Bắt buộc
	Phân loại: Trung bình
Mô tả tóm tắt: Người chơi bắt đầu lượt chơi.	
Các mối quan hệ: + Association (kết hợp): Người chơi.	
Luồng xử lý bình thường của sự kiện: <ol style="list-style-type: none"> Ở trang chủ, hệ thống sẽ mặc định các cài đặt như sau: <ul style="list-style-type: none"> - Chế độ: words - Config: 10 words - Độ khó từ vựng: English Người chơi có thể thay đổi các cài đặt trên. Người chơi bắt đầu đánh bất kỳ ký tự nào trên bàn phím để bắt đầu lượt chơi. Hệ thống sẽ bắt đầu tính thời gian. Lượt chơi kết thúc khi hết thời gian hoặc hết ký tự trên màn hình. Hệ thống cập nhật kết quả vào bảng Record. 	
Luồng luân phiên/đặc biệt (Alternate/Exceptional flows):	

3.9. Reset lượt chơi

Tên Use case: Reset lượt chơi	
Actor chính: Người chơi	Mức độ cần thiết: Bắt buộc
	Phân loại: Trung bình
Mô tả tóm tắt: Người chơi bắt đầu lại lượt chơi.	
Các mối quan hệ: + Association (kết hợp): Người chơi.	
Luồng xử lý bình thường của sự kiện: <ol style="list-style-type: none"> Sau khi kết thúc lượt hoặc đang trong lượt chơi, người chơi click nút “Reset” hoặc Tab + Enter để reset lượt chơi. Lượt chơi mới sẽ bao gồm các cài đặt từ lượt chơi trước. 	
Luồng luân phiên/đặc biệt (Alternate/Exceptional flows):	

3.10. Xem kết quả

Tên Use case: Xem kết quả	
Actor chính: Người chơi	Mức độ cần thiết: Bắt buộc
	Phân loại: Trung bình

Mô tả tóm tắt: Người chơi xem kết quả sau lượt chơi.
Các mối quan hệ: + Association (kết hợp): Người chơi. + Includes (bao gồm): Hiện thị kết quả
Luồng xử lý bình thường của sự kiện: 1. Sau khi hoàn thành lượt chơi, người chơi xem kết quả. 2. Các kết quả bao gồm: - WPM và độ chính xác. - Biểu đồ WPM mỗi giây. - Độ khó từ vựng. - Chế độ và config. - Số ký tự đánh. - Thời gian hoàn thành lượt chơi.
Luồng luân phiên/đặc biệt (Alternate/Exceptional flows):

3.11. Hiện thị thống kê kết quả

Tên Use case: Hiện thị thống kê kết quả	
Actor chính: Hệ thống	Mức độ cần thiết: Bắt buộc
	Phân loại: Trung bình
Mô tả tóm tắt: Hệ thống hiển thị thống kê kết quả.	
Các mối quan hệ: + Association (kết hợp): Hệ thống	
Luồng xử lý bình thường của sự kiện: 1. Hệ thống truy xuất các thông tin sau từ cơ sở dữ liệu MySQL dựa vào ID người chơi: - Số lượt chơi hoàn thành, WPM trung bình, tổng thời gian chơi. - Các lượt chơi tốt nhất sắp xếp theo các config của chế độ “Time”. - Các lượt chơi tốt nhất sắp xếp theo các config của chế độ “Words”. - Các lượt chơi tốt nhất sắp xếp theo các config của chế độ “Quote”. - Tất cả WPM. - Lịch sử lượt chơi. - Ngày tạo tài khoản. 2. Hệ thống hiển thị các thông tin vừa truy xuất.	
Luồng luân phiên/đặc biệt (Alternate/Exceptional flows):	

3.12. Kiểm tra CapLock

Tên Use case: Kiểm tra CapLock	
Actor chính: Hệ thống	Mức độ cần thiết: Bắt buộc
	Phân loại: Trung bình
Mô tả tóm tắt: Hệ thống kiểm tra nếu phím CapLock đang hoạt động.	
Các mối quan hệ: + Association (kết hợp): Hệ thống	
Luồng xử lý bình thường của sự kiện: 1. Hệ thống kiểm tra sự kiện “KeyDown” của người dùng. 2. Nếu người dùng bấm phím CapLock, hệ thống hiện thông báo “CapLock”.	

3. Hệ thống tắt thông báo khi phím “CapLock” không còn hoạt động.
4. Hệ thống tiếp tục kiểm tra sự kiện “KeyDown” của người dùng.

Luồng luân phiên/đặc biệt (Alternate/Exceptional flows):

3.13. Tính toán kết quả

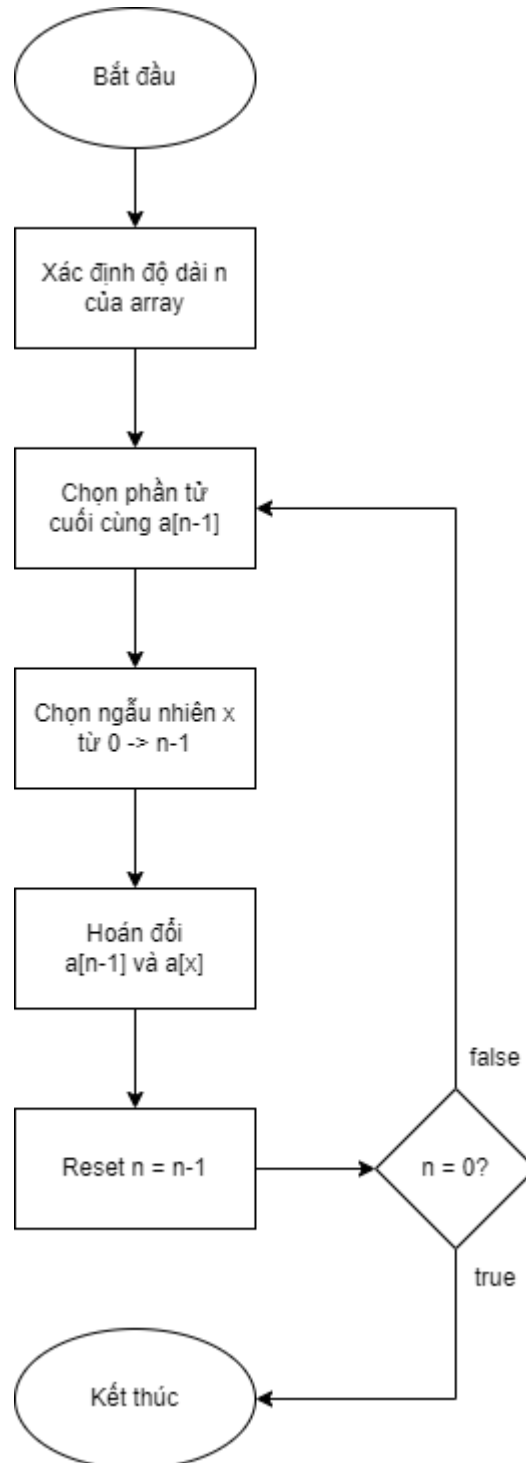
Tên Use case: Tính toán kết quả	
Actor chính: Hệ thống	Mức độ cần thiết: Bắt buộc
	Phân loại: Trung bình
Mô tả tóm tắt: Hệ thống tính toán kết quả sau khi kết thúc lượt chơi.	
Các mối quan hệ:	
+ Association (kết hợp): Hệ thống	
Luồng xử lý bình thường của sự kiện:	
<ol style="list-style-type: none"> 1. Hệ thống tính độ chính xác. 2. Hệ thống tính WPM mỗi giây. 3. Hệ thống tính WPM tổng quan với giải thuật Levenshtein Distance. 	
Luồng luân phiên/đặc biệt (Alternate/Exceptional flows):	

3.14. Hiện thị kết quả

Tên Use case: Hiện thị kết quả	
Actor chính: Hệ thống	Mức độ cần thiết: Bắt buộc
	Phân loại: Trung bình
Mô tả tóm tắt: Hệ thống hiển thị các kết quả vừa tính được.	
Các mối quan hệ:	
+ Association (kết hợp): Hệ thống	
+ Inlucdes (bao gồm): Tính toán kết quả	
Luồng xử lý bình thường của sự kiện:	
<ol style="list-style-type: none"> 1. Hệ thống cập nhật biểu đồ WPM mỗi giây với dữ liệu vừa tính được. 2. Hệ thống hiển thị WPM tổng quan và độ chính xác. 3. Hệ thống hiển thị các cài đặt của lượt chơi. 4. Hệ thống hiển thị tổng ký tự đánh và thời gian hoàn thành. 	
Luồng luân phiên/đặc biệt (Alternate/Exceptional flows):	

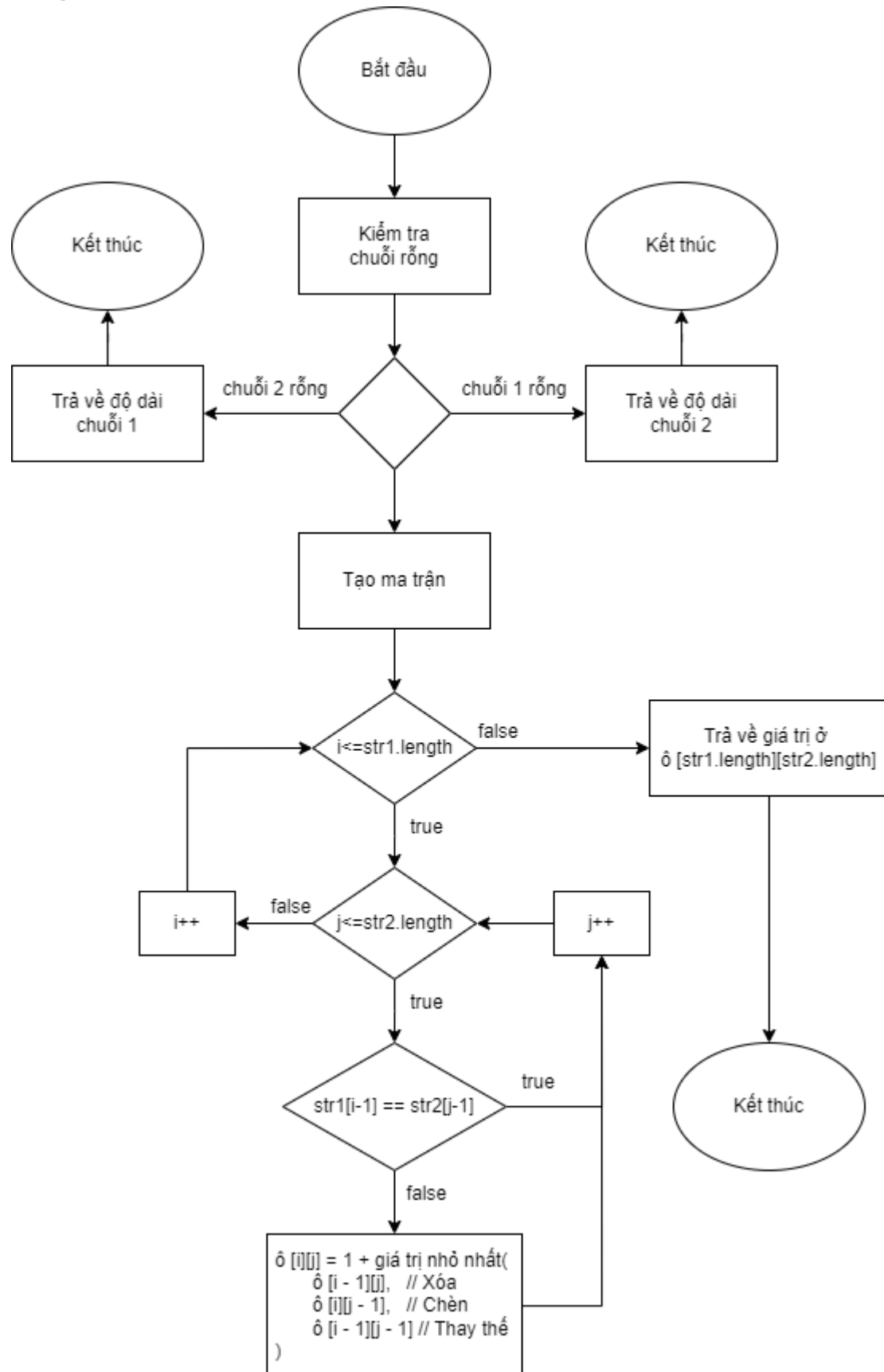
4. Lưu đồ giải thuật

4.1. Thuật toán xáo trộn Knuth



Hình 4 - Lưu đồ giải thuật xáo trộn Knuth

4.2. Thuật toán Levenshtein Distance



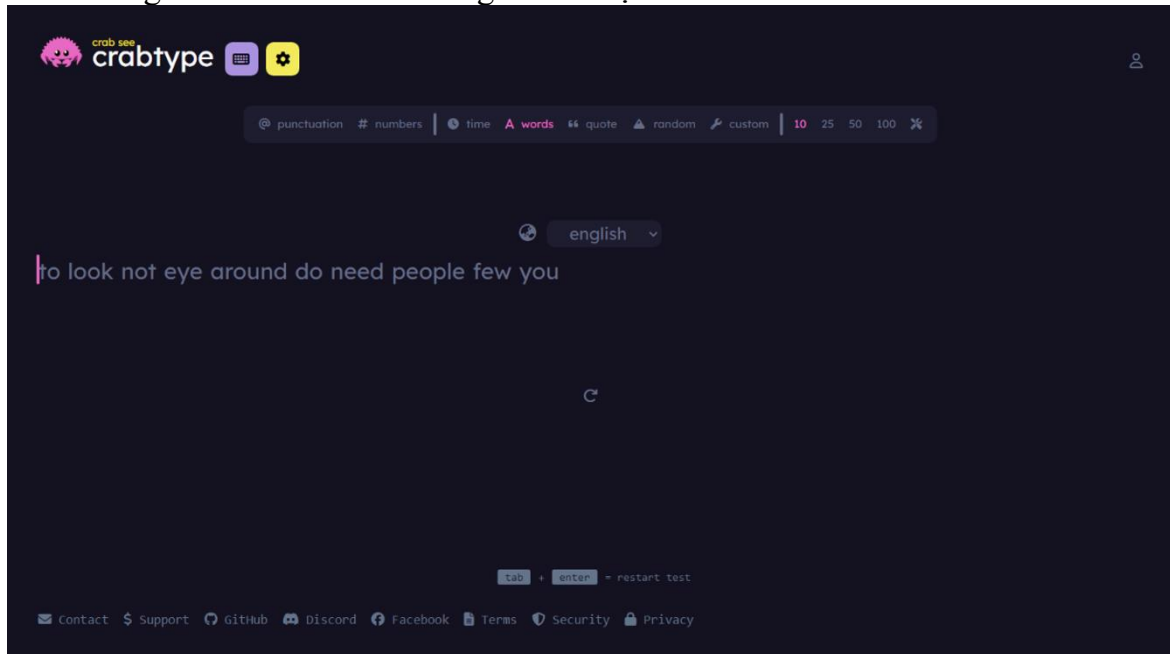
Hình 5 - Lưu đồ giải thuật Levenshtein Distance

5. Giới thiệu chương trình

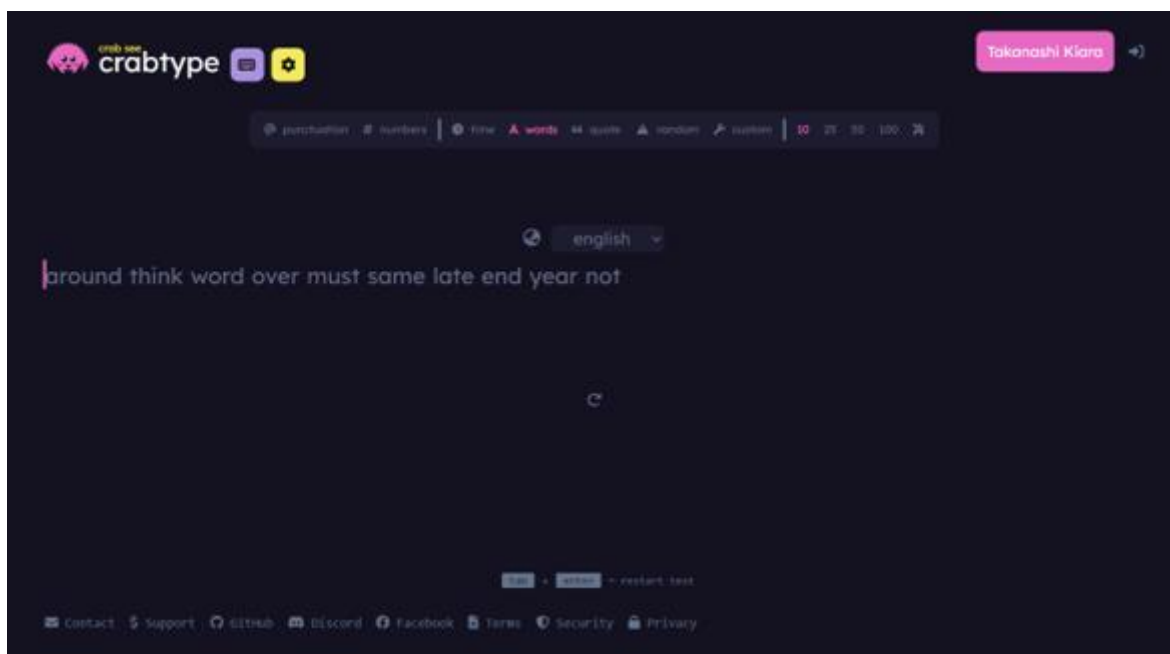
5.1. Trang chủ

Khi người dùng truy cập website, trang chủ sẽ bao gồm các thành phần sau đây:

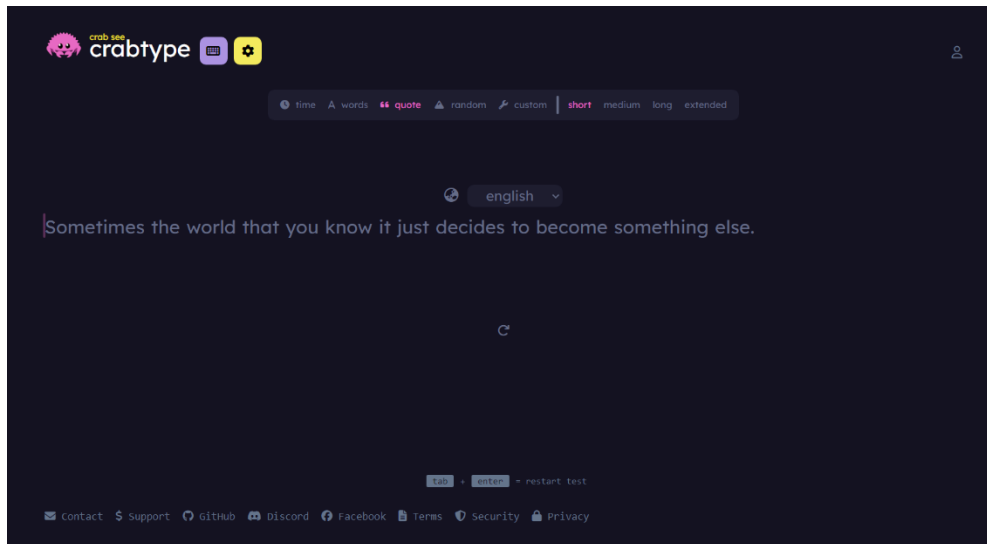
- Header bao gồm logo website bên trái và nút “Xác thực tài khoản” bên phải.
- Thanh cài đặt bao gồm các chế độ và config.
- Combo box chứa các độ khó từ vựng.
- Khu vực các ký tự hiển thị
- Nút reset lượt chơi.
- Footer gồm danh sách các thông tin liên lạc.



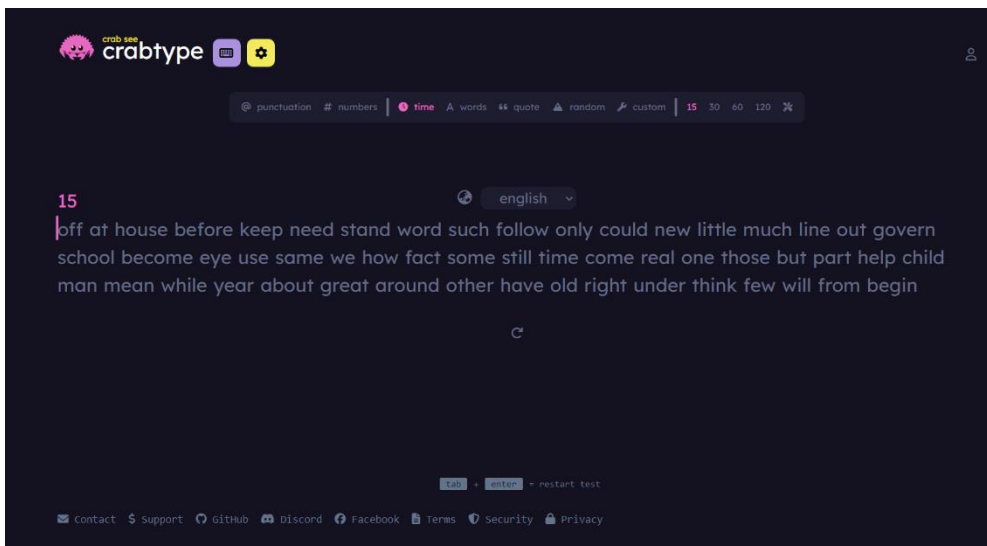
Hình 6 - Trang chủ



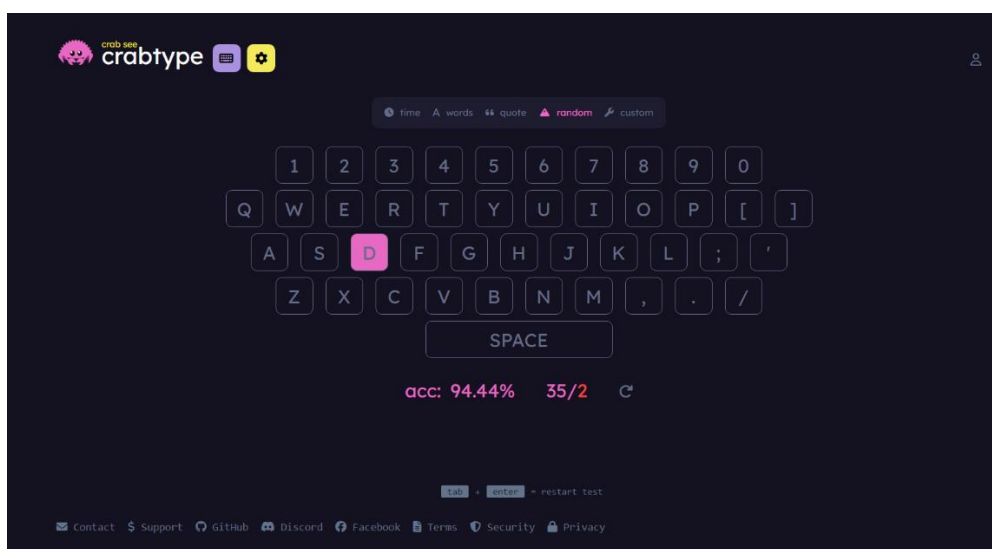
Hình 7 - Trang chủ sau khi xác thực tài khoản



Hình 8 - Trang chủ (chế độ quote)



Hình 9 - Trang chủ (chế độ time)



Hình 10 - Trang chủ (chế độ random)

5.2. Trang cá nhân / tài khoản



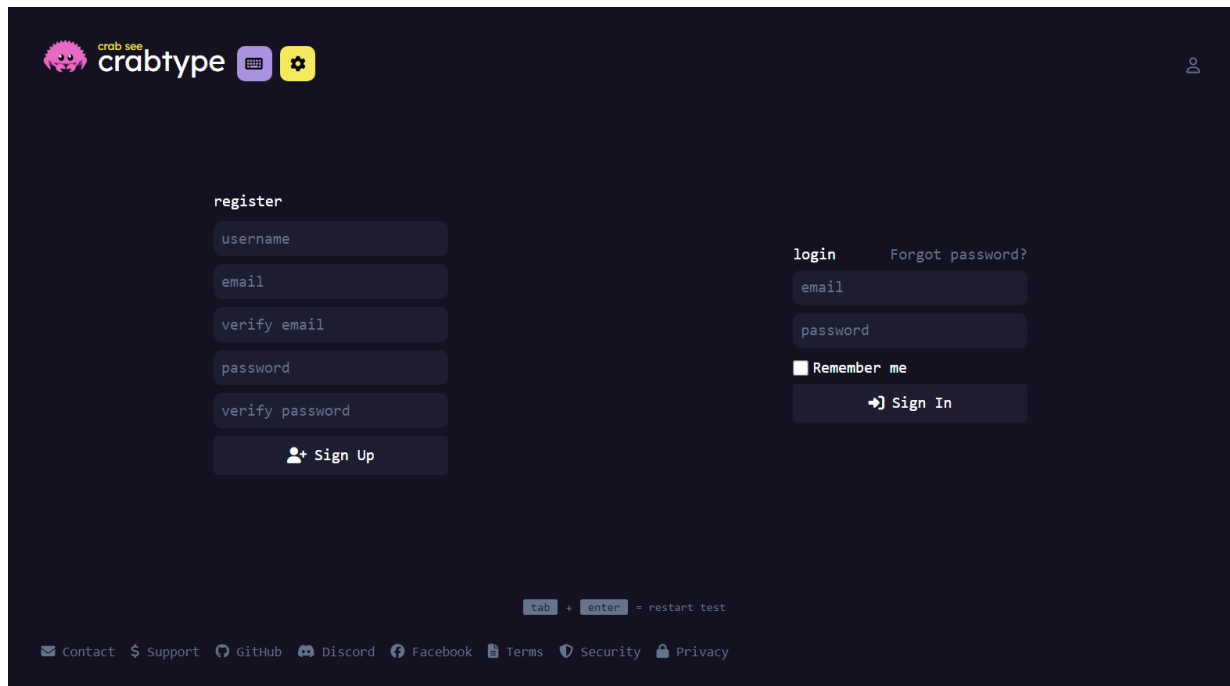
Hình 11 - Trang cá nhân / tài khoản

Người chơi có thể truy cập trang cá nhân sau khi xác thực tài khoản, nó bao gồm các thành phần sau đây:

- Các thông tin tài khoản: Tên người chơi, ngày tạo tài khoản, lượt chơi hoàn thành, WPM trung bình, tổng thời gian chơi.

- WPM cao nhất và độ chính xác của từng config trong chế độ time.
- WPM cao nhất và độ chính xác của từng config trong chế độ words.
- WPM cao nhất và độ chính xác của từng config trong chế độ quote.
- Biểu đồ cột thống kê tất cả các WPM của người chơi.
- Lịch sử kết quả lượt chơi.

5.3. Trang xác thực

The image shows the Crabtype website's authentication page. At the top left is the 'crabtype' logo with a pink crab icon. To its right are icons for a keyboard and a gear. In the top right corner is a user profile icon. The main content area is divided into two sections: 'register' on the left and 'login' on the right. The 'register' section contains input fields for 'username', 'email', 'verify email', 'password', and 'verify password', followed by a 'Sign Up' button with a user icon. The 'login' section contains input fields for 'email' and 'password', a 'Remember me' checkbox, and a 'Sign In' button with a right arrow icon. A link 'Forgot password?' is located above the password field. Below the login fields is a button that says 'tab + enter = restart test'. At the bottom of the page is a footer with links for 'Contact', 'Support', 'Github', 'Discord', 'Facebook', 'Terms', 'Security', and 'Privacy'.

Hình 12 - Trang xác thực

Trang xác thực gồm hai thành phần chính:

- Bên trái là biểu mẫu đăng ký tài khoản: Người chơi điền các thông tin username, email và mật khẩu, sau đó click “Sign Up” để đăng ký. Nếu email đã được sử dụng thì website sẽ báo lỗi.
- Bên phải là biểu mẫu đăng nhập tài khoản: Người chơi nhập email và mật khẩu, sau đó click “Sign In” để đăng nhập. Nếu người chơi nhập email hoặc mật khẩu sai, website sẽ báo lỗi.

PHẦN 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết quả đạt được

- Hiểu được logic xử lý của thuật toán xáo trộn Knuth và Levenshtein Distance. Áp dụng được hai thuật toán vào các chức năng của website.
- Hiểu được cơ bản về cách thức xây dựng và hoạt động của một website full-stack.
- Sử dụng thành công framework React.js trong việc xử lý front-end.
- Sử dụng thành công framework Express.js trong xây dựng các APIs.
- Sử dụng thành công cơ sở dữ liệu MySQL để lưu trữ dữ liệu liên quan.
- Thiết kế của website thân thiện và dễ sử dụng.
- Đề tài “Trò chơi đánh máy tốc độ CrabType” đã được hoàn thiện với các chức năng được đưa ra.

2. Hạn chế

- Tính năng bảo mật cần được phát triển hơn.
- Website chỉ hỗ trợ ngôn ngữ Anh.
- Mã nguồn chưa hoàn toàn bám theo kiến trúc Component-based.
- Mã nguồn còn khó đọc do thiếu kinh nghiệm.
- Một số kiến thức của React.js chưa được sử dụng đúng cách.
- Con trỏ đôi khi không xuất hiện đúng vị trí đối với màn hình lớn.

3. Hướng phát triển

- Phát triển thêm về độ khó chung thay vì chỉ độ khó về từ vựng.
- Thêm token vào chức năng xác thực tài khoản để tăng cường bảo mật.
- Thêm ngôn ngữ hỗ trợ mới.
- Thêm chức năng lọc lịch sử lượt chơi trong trang cá nhân.
- Thêm chức năng xuất file excel của lịch sử lượt chơi.
- Thêm chức năng thực hiện lại lượt chơi trước.
- Thêm chức năng chụp màn hình kết quả lượt chơi.

TÀI LIỆU THAM KHẢO

- [1] **Jack Miodec**, <https://monkeytype.com/>, 14/05/2024.
- [2] **Hrishitva Patel, Goutham Ravichandran**, *Levenshtein Algorithm*, <https://hcommons.org/deposits/objects/hc:46164/datastreams/CONTENT/content>, 14/05/2024.
- [3] **Chandra Kirana* , Benny Wijaya, Abdul Holil**, *Implementation of the Fisher-Yates Shuffle Algorithm in Exam-Problem Randomization on M-Learning Applications*, <https://pdfs.semanticscholar.org/6c7a/0ff587576cd1a0517451ba5e35c56ed21a58.pdf>, 14/05/2024.
- [4] **Matt Groeber**, *How to Calculate Typing Speed (WPM) and Accuracy*, <https://www.speedtypingonline.com/typing-equations#:~:text=To%20calculate%20Net%20WPM%2C%20take,known%20as%20the%20error%20rate.>, 14/05/2024.
- [5] Tailwind CSS, <https://tailwindcss.com/docs/installation>, 14/05/2024.
- [6] React.js, <https://react.dev>, 14/05/2024.
- [7] Express.js, <https://expressjs.com>, 14/05/2024.
- [8] Axios, <https://axios-http.com/docs/intro>, 14/05/2024.
- [9] Animate.css, <https://axios-http.com/docs/intro>, 14/05/2024.
- [10] **Jeremy Ayerst**, <https://github.com/reactchartjs/react-chartjs-2>, 14/05/2024.
- [11] **Dawid Paszko**, https://youtu.be/E_tZH9R_zi8?si=vxn0f5BFlqsth0mO, 14/05/2024.
- [12] **Muyang Guo**, <https://github.com/gamer-ai/eletypes-frontend>, 14/05/2024.